Abstract

In this study, we tackled the reduction of computational complexity by pruning the _igo_ game tree using the potential model based on the knowledge expression of _igo_. The potential model considers _go_ stones as potentials. Specific potential distributions on the _go_ board result from each arrangement of the stones on the _go_ board. Pruning using the potential model categorizes the legal moves into effective and ineffective moves in accordance with the threshold of the potential. In this experiment, 4 kinds of pruning strategies were evaluated. The best pruning strategy resulted in an 18% reduction of the computational complexity, and the proper combination of two pruning methods resulted in a 23% reduction of the computational complexity. In this research we have successfully demonstrated pruning using the potential model for reducing computational complexity of the _go_ game.

**Keyword:** Monte-Carlo Method ; Game Tree ; Pruning ; _Igo_ ; Potential

1. Introduction

_Monte-Carlo go_ [1] is the computer _igo_ that satisfies strength without the knowledge expressions of _igo_, and it has attracted much attention recently. _Monte-Carlo go_ is the computer _igo_ which applies the Monte-Carlo method [2] to the game tree of _igo_. _Monte-Carlo go_ is very computationally intensive. However, reduction of the computational complexity is possible by properly pruning the _igo_ game tree.

Previous research of computer _igo_ placed emphasis on improving evaluation functions through the expert implementation of the knowledge expressions of _igo_. However, since the advantage of _Monte-Carlo go_ was confirmed, research of _Monte-Carlo go_ has become the mainstream in place of previous research.

It is more difficult to make evaluation functions in _igo_ compared to other two-player, zero-sum, logical perfection information games (such as chess and _shogi_). This is a major reason why the strength of computer _igo_ remained low. On the contrary, _Monte-Carlo go_ can strategize without evaluation functions. As a result, this characteristic generates high strengths of computer _igo_. Several computer _igo_ developed from _Monte-Carlo go_ [3, 4] have shown
their high strengths by winning major computer *igo* tournaments. Accordingly, current studies of computer *igo* are being concentrated on high-efficiency technology of Monte-Carlo *go*.

Monte-Carlo *go* is very computationally intensive. Efficiencies of tree search are very important. There are several studies based on probabilistic analyses for efficiencies of tree search [5]. On the other hand, there are few studies based on the knowledge expressions of *igo*. The potential model is one of the knowledge expressions of *igo*. Thus, there is little information on the effects of the potential model for pruning in Monte-Carlo *go*.

The purpose of this study is to examine reduction effects of the calculation load required by Monte-Carlo *go* by the pruning using the potential model, and to evaluate the usefulness of the potential model in *igo*.

We implemented functions of the pruning by the potential model in Monte-Carlo *go*. The potential model is to consider *go* stones as potentials. Specific potential distributions and potential gradient distributions on the *go* board result from each arrangement of the stones on the *go* board. Pruning using the potential model categorizes the legal moves into effective and ineffective moves in accordance with the threshold for the potential or the potential gradient. This is the “Potential Model Pruning” in this study. We have called these implemented functions “Potential Filters.”

In this experiment, 4 kinds of Potential Filters were evaluated, and another pruning strategy, pruning randomly (Random Filter), was also evaluated for comparison purposes to the 4 kinds of Potential Filters.

These Filters were evaluated by measuring the following experimental items:

i. Variations of strength of Monte-Carlo *go* by Filters (and combination of two Potential Filters.)
ii. Compatibilities among the effective moves evaluated by Monte-Carlo search and the other effective moves evaluated by Potential Filters.

In this research we have successfully demonstrated pruning using the potential model for reducing computational complexity of *igo* and shown the usefulness of the potential model in *igo*. For our future research, we intend to expand the proposed strategy to tackle more complex games.

2. Monte-Carlo *Go*

Monte-Carlo *go* is computer *igo* that evaluates legal moves at each phase to choose the next move by simulation based on the Monte-Carlo method (Monte-Carlo search.)

Monte-Carlo search consists of many moves of a simulation. This simulation is called “Play Out.” Play Out involves both sides constantly choosing the next move alternately and randomly from the current phase to the end game. Play Out calculates an estimation ($\bar{X}_i$) for each legal move ($i$). $s_i$ is the number of times of Play Out. $X_i$ is the total considerations of Play Out. In Play Out, if the offensive wins, the consideration is 1, and if it loses, the consideration is 0.

$$\bar{X}_i = \frac{X_i}{s_i} \quad (1)$$

3. Potential Model

In *igo*, stones raise the possibility of the surrounding intersection becoming their territory. A Potential is the formulated variation of possibility. There are several researches using potential for evaluation of phases [6, 7]. Potential nears human recognition more than Monte-Carlo search and is the knowledge expression of *igo*. In their researches, a center of a stone has a potential "extremum" and potential is decreased according to distance from the center.

3.1. Definition of Potential

The definition of potential in this experiment is shown in Formula (2, 3, 4). Black *go* stones were assumed to have positive potential and white *go* stones were assumed to have negative potential. Thus specific potential
distributions on the go board result from each arrangement of the stones on the go board. An example of a potential distribution is shown in Fig. 1.

$$P(X, Y) = \sum_{k=1}^{n} P_k(X, Y)$$  \hspace{1cm} (2)

$$P_i(X, Y) = \frac{\text{Stone}_i}{2d}$$ \hspace{1cm} (3)

$$d = \sqrt{(X - x_i)^2 + (Y - y_i)^2}$$ \hspace{1cm} (4)

3.2. Definition of Potential Gradient

The definition of potential gradient in this experiment is shown in Fig. 3 and Formula (5). $PG(i, j)$ is a potential gradient at an intersection. An example of a potential distribution is shown in Fig. 2.

3.3. Pruning Using Potential Model

3.3.1. Function of Potential Filters

Potential Filters are pruning instruments in this experiment. At each phase to choose the next move, these Filters pruned legal moves according to the following procedures:

i. Calculate potential distribution result from arrangement of go stones on the go board.

ii. Rank legal moves by each magnitude of potential (or potential gradient.)

iii. Categorize the ranked legal moves into effective and ineffective moves in accordance with thresholds for the ranking. Each Potential Filter has unique threshold levels.

iv. Eliminate the ineffective moves from candidates for the next move (Run Monte-Carlo search only on effective moves.)

For comparison, the Random Filter categorizes legal moves randomly regardless of potential distribution. In accordance with the number of eliminated legal moves, the computation load of Monte-Carlo search is reduced.

3.3.2. Configurations of Potential Filters

Table 1 shows the threshold conditions of the 5 kinds of Filters. Each Potential Filter ranked legal moves in descending order of potential or potential gradient values, and categorized in accordance with each threshold condition for the ranking.

All Filters mutually reduced by half the number of legal moves. Thus all Filters reduced by half the computational load at each phase to choose the next move.

Fig. 4 shows image of each Potential Filter. Black shows positive and white shows negative. Figures indicate the rank of the intersection. Shaded areas show ineffective and pruned moves.
### Table 1. Types of Pruning Filters

<table>
<thead>
<tr>
<th>Filter</th>
<th>Targets of Ranking</th>
<th>Eliminated Legal Moves</th>
</tr>
</thead>
<tbody>
<tr>
<td>Potential Filter 1</td>
<td>Potential</td>
<td>Low 50% of ranks</td>
</tr>
<tr>
<td>Potential Filter 2</td>
<td>Potential</td>
<td>Top 50% of ranks</td>
</tr>
<tr>
<td>Potential Filter 3</td>
<td>Potential</td>
<td>Above 25% and below 75% of ranks</td>
</tr>
<tr>
<td>Potential Filter 4</td>
<td>Potential Gradient</td>
<td>Low 50% of ranks</td>
</tr>
<tr>
<td>Random Filter</td>
<td>–</td>
<td>Legal moves eliminated 50% of the time</td>
</tr>
</tbody>
</table>

![Fig. 4 Examples of Pruning](image)

#### 3.3.3. On and Off Switch of Potential Filter

Potential Filters had a switching point, which switched their states ON and OFF. This switching point was within a range of legal intersection numbers on the go board. In this experiment, a switching point was selectable from 1 to 81 because the board size was 9x9 (= 81).

During the course of a game, in the case a remaining legal move number on the go board was above a switching point, the Potential Filters were ON. If a remaining legal move number was under a switching point, the Potential Filters were OFF. In this experiment, boundaries where Potential Filters became ineffective from effective were measured by changing the switching point. The boundaries were the points where winning percentages crossed a 57% threshold *(4. Strength of Monte-Carlo Go with Potential Filters.)*

Monte-Carlo search has higher performance when a game tree is small. In contrast, Monte-Carlo search has low performance when a game tree is large. Thus, pruning is effective in the opening game. However, afterwards, pruning gradually becomes ineffective.

#### 4. Strength of Monte-Carlo Go with Potential Filters

The strength of Monte-Carlo go with Potential Filters was indicated by its winning percentage against normal Monte-Carlo go. Monte-Carlo go with Potential Filters used the initiative move while normal Monte-Carlo go used the passive move. In a match-up between two normal Monte-Carlo go, the winning percentage of the initiative move was 57%. *(The winning percentage of initiative exceeded 50% because the initiative move was advantageous.)* Therefore, 57% is considered the average level of normal strength.

#### 5. Compatibilities

The definition of the compatibilities between the effective moves evaluated by Monte-Carlo search and the other effective moves evaluated by Potential Filters is shown in *(Formula 6).*

\[
(\text{Compatibility}) = \frac{\text{Num}(EMove_m \cap EMove_p)}{\text{Num}(Legal\ Move)/2} \quad (6)
\]

<table>
<thead>
<tr>
<th>( EMove_m )</th>
<th>The top 50 percent of legal moves evaluated by Monte-Carlo search</th>
</tr>
</thead>
<tbody>
<tr>
<td>( EMove_p )</td>
<td>Effective moves evaluated by Potential Filters</td>
</tr>
</tbody>
</table>

#### 6. Experiment Environment

In our experiments, the go board size was set at 9x9 and the maximum Play Out number was set at 100.
7. Results and Observations

The strength of Monte-Carlo go with Potential Filters is shown in Fig. 5, left axis. Strength transitioned with Filters and switching points. A winning percentage of 57% and calculating the results of the Random Filter were important for comparing and evaluating the effects and tendency of Potential Filters. Total Play Out numbers needed in one game are shown in Fig. 5, right scale. Total Play Out numbers transitioned with Filters and switching points.

Compatibilities between the effective moves evaluated by Monte-Carlo search and the other effective moves evaluated by Potential Filters are shown in Fig. 6.

<table>
<thead>
<tr>
<th></th>
<th>Max</th>
<th>Min</th>
<th>Border</th>
<th>Play Out Number</th>
<th>Reduction Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>56.4</td>
<td>12.6</td>
<td>–</td>
<td>332000</td>
<td>0.0</td>
</tr>
<tr>
<td>Potential Filter 1</td>
<td>62.2</td>
<td>00.0</td>
<td>77</td>
<td>316300</td>
<td>04.7</td>
</tr>
<tr>
<td>Potential Filter 2</td>
<td>63.1</td>
<td>00.0</td>
<td>73</td>
<td>301400</td>
<td>09.2</td>
</tr>
<tr>
<td>Potential Filter 3</td>
<td>53.3</td>
<td>00.0</td>
<td>–</td>
<td>332000</td>
<td>00.0</td>
</tr>
<tr>
<td>Potential Filter 4</td>
<td>66.4</td>
<td>00.0</td>
<td>64</td>
<td>270800</td>
<td>18.4</td>
</tr>
<tr>
<td>Combination</td>
<td>66.4</td>
<td>00.0</td>
<td>59</td>
<td>255500</td>
<td>23.0</td>
</tr>
</tbody>
</table>

Fig. 5 Strengths of Monte-Carlo Go with Potential Filters

In theory, when the Random Filter was used, the next move became the best move by Monte-Carlo search 50% of the time and the second or several moves thereafter by Monte-Carlo search the other 50% of the time. This compatibility with Monte-Carlo go was consistently 50%.

In the case that the number of legal moves was large, Monte-Carlo search had low precision. Thus, there was no big decrease of strength, because there was no defining difference between the best move and the second or several
moves thereafter by Monte-Carlo search. The precision of Monte-Carlo search increased with a decrease in the number of legal moves. The strength of the Random Filter decreased gradually with a decrease in the number of legal moves. However, even if the Random Filter was used at all phases, the winning percentage stopped falling around 15% because the second or several moves thereafter were not entirely wrong moves.

Potential Filter 1 became the bias around which black stones gathered. In the opening game, these collective black stones effectively strengthened initiative territory. However, in the middle game, the initiative move could not expand its territory. As a result, the passive move acquired more territory than the initiative move on the go board. When strength exceeded 57%, Potential Filter 1 properly pruned ineffective moves that Monte-Carlo search was unable to do. This compatibility with Monte-Carlo go was consistently about 50%.

Potential Filter 2 became the bias where black stones were attracted around white stones. In the opening game, black stones effectively suppressed white stones. As a result, the passive move acquired more territory than the initiative move on the go board. When strength exceeded 57%, Potential Filter 2 properly pruned ineffective moves. This compatibility with Monte-Carlo go was consistently about 50%.

Potential Filter 3 became the bias where black stones were scattered on the go board. These black stones were removed easily by collective white stones. As a result, the passive move acquired more territory than the initiative move on the go board. In the opening game, Potential Filter 3 barely pruned ineffective moves. However, this was the only compatibility with Monte-Carlo go that increased in the middle game.

Potential Filter 4 became the bias where black stones were attracted around black and white stones, and areas between black and white stones were closed. This is important in igo. Potential Filter 4 could prune more properly than the other Filters. Only this compatibility with Monte-Carlo go was above 50% in the opening game. However, it decreased gradually during the course of the game.

As for Combination, Potential Filter 4 and Potential Filter 3 combined pruned a game tree more properly than Potential Filter 4 alone.

8. Summary

In this study, we tackled the reduction of computational complexity by pruning the igo game tree using the potential model based on the knowledge expression of igo. In our experiments, 4 kinds of pruning strategies (Potential Filters) were evaluated for their removal effect. Maintaining normal strength of Monte-Carlo go, Potential Filter 4, based on potential gradient distribution, could reduce up to 18% of total Play Out numbers needed in one game. Potential gradient could identify important areas around black and white stones as well as close areas between black and white stones. In addition, combining Potential Filter 4 and Potential Filter 3 could reduce up to 23% of total Play Out numbers by switching Potential Filter 4 to Potential Filter 3 at the point where the strength of Potential Filter 4 began to decline (switching point 68). This shows the tendency of igo transition as the game progresses.

In this research we successfully demonstrated pruning using the potential model for reducing computational complexity of the go game. However, our experiments were limited as the Play Out number was set at 100 and the board size was set at 9x9. For our future research, we intend to expand the proposed strategy to tackle more complex games with larger Play Out numbers and go board size.

Reference